COMP 354
Introduction to Software Engineering

Greg Butler

Computer Science and Software Engineering
Concordia University, Montreal, Canada

Office: EV 3.219 Email: `gregb@cs.concordia.ca`

Winter 2015

Course Web Site:

http://users.encs.concordia.ca/˜gregb/home/comp354-w2015.html

Introduction to Software Engineering

Software engineering

Area of computer science concerned with building large software systems

Challenge

Tremendous advances in hardware have not been accompanied by comparable advances in software

Introduction to Software Engineering

What is Software Engineering?

The process of **solving customers problems** by the systematic development and evolution of large, high-quality software systems within cost, time and other constraints.

Solving customers' problems

This is the goal of software engineering
Sometimes the solution is to buy, not build
Adding unnecessary features does not help solve the problem
Software engineers must communicate effectively to identify and understand the problem

What is Software Engineering?

Systematic development and evolution

An engineering process involves applying well understood techniques in a organized and disciplined way
Many well-accepted practices have been formally standardized
Most development work is evolution

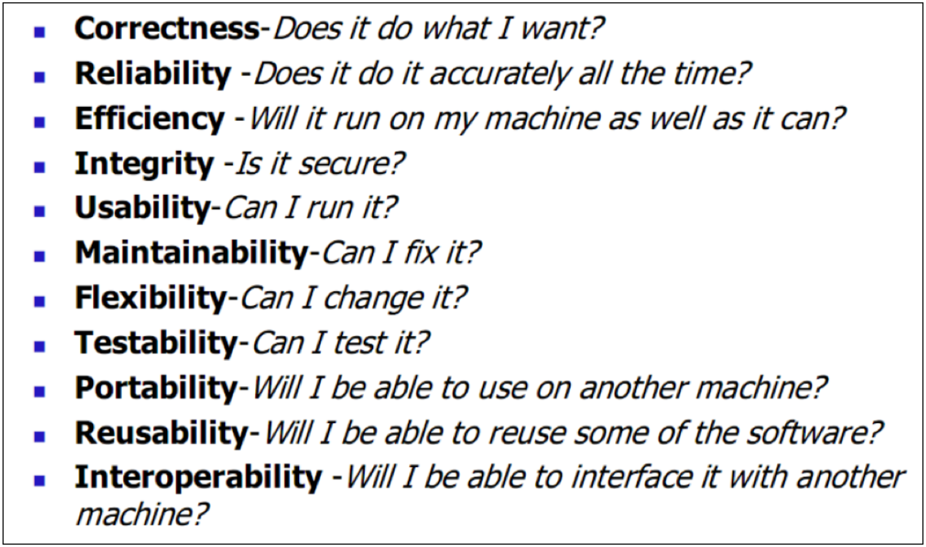
Large, high quality software systems

Software engineering techniques are needed because large systems cannot be completely understood by one person
Teamwork and co-ordination are required
The end-product that is produced must be of sufficient quality

Cost, time and other constraints

Finite resources
The benefit must outweigh the cost
Others are competing to do the job cheaper and faster
Inaccurate estimates of cost and time have caused many project failures

Additional Software Quality Attributes



- **Correctness**-*Does it do what I want?*
- **Reliability** -*Does it do it accurately all the time?*
- **Efficiency** -*Will it run on my machine as well as it can?*
- **Integrity** -*Is it secure?*
- **Usability**-*Can I run it?*
- **Maintainability**-*Can I fix it?*
- **Flexibility**-*Can I change it?*
- **Testability**-*Can I test it?*
- **Portability**-*Will I be able to use on another machine?*
- **Reusability**-*Will I be able to reuse some of the software?*
- **Interoperability** -*Will I be able to interface it with another machine?*

Stakeholders

Customer
solves problems at an acceptable cost in terms of money paid and resources used

User
easy to learn;
efficient to use;
helps get work done

Developer
easy to design;
easy to maintain;
easy to reuse its parts

Development Manager
sells more and pleases customers while costing less to develop and maintain

Software Processes

A software process prescribes all major activities involved in building a software system

Uses resources, within a set of constraints, to produce intermediate and final products

May be composed of sub-processes

Activities are organized in a sequence

Has a set of guiding principles to explain goals

Software Development Stages

Requirements Analysis & Specification

Conceptual/System Design

Detailed/Program Design

Implementation/Coding

Unit & Integration Testing

System Testing

System Delivery and Deployment

Maintenance

Sequential vs. Iterative Development Process

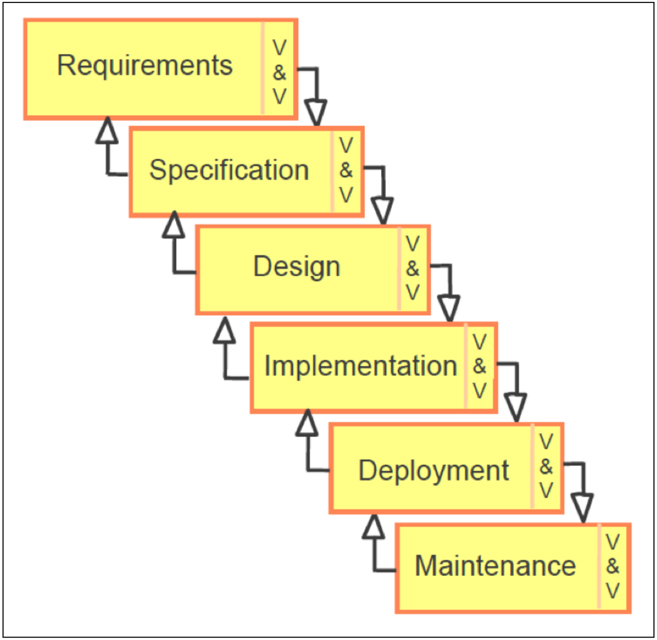
Sequential Development

Development is organized into a *sequence of development phases*. The termination of one phase enables a subsequence phase.

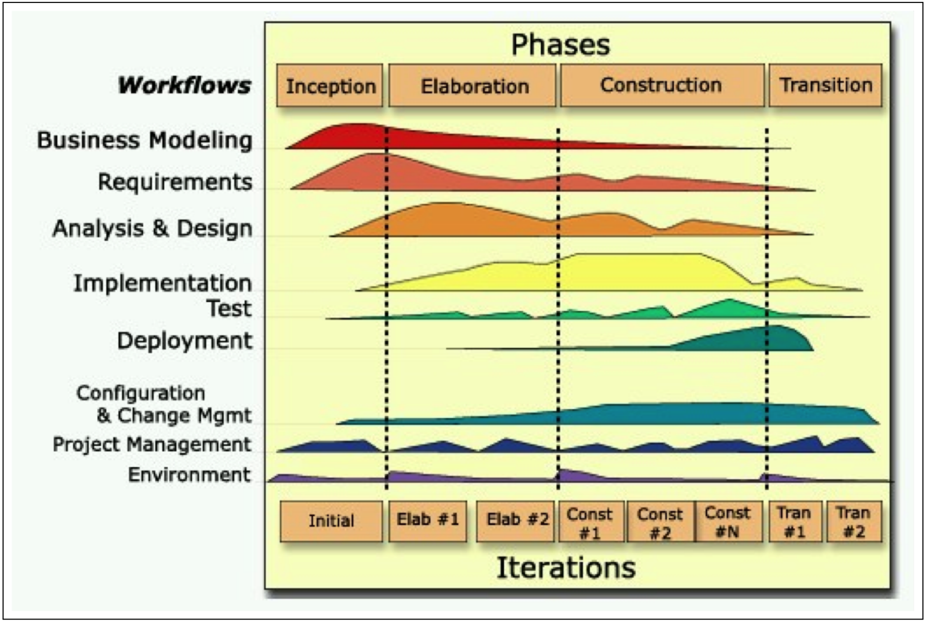
Iterative Development

Development is organized into a *series of* short, fixed-length mini-projects called *iterations*; the outcome of each is a tested, integrated, and executable system. [Craig Larman, 2002]

Example of a Software Process: Waterfall Model



Example of a Software Process: The Unified Process



In SE, Change is constant

Change ... at a rapid pace

This means you need to be an agile learner

How rapidly have things evolved ...?

Computing “Predictions” 1943–2010

Original quotes from LICS 2006 keynote speech of Dana Scott (Carnegie Mellon University) on *“The Future of Proof”*.

Predictions

Thomas J. Watson (1943)

"I think there is a world market for maybe five computers."

Popular Mechanics (1949)

forecasting the relentless march of the machine
"Computers in the future may weigh no more than 1.5 tons."

Prentice-Hall business editor (1957)

"I have traveled the length and breadth of this country and talked with the best people, and I can assure you that data processing is a fad that won't last out the year."

IBM engineer (1968)

commenting on the microchip
"But what ... is it good for?"

Software Runs The Modern World

Software touches almost every aspect of our modern world

Often possible to combine other passions with SE

Arts: dance, music, ...;

Astronomy

Biology: bioinformatics (CU M$+ research)

Communications

...

Surgery

...

Zoology

Including applications you might never have imagined